

09114750 A



(19)

Generated Document.

(11) Publication number: **09114'****PATENT ABSTRACTS OF JAPAN**

(51) Intl. Cl.: G06F 13/00

(21) Application number: **07265586**(22) Application date: **13.10.95**

(30) Priority:

(43) Date of application publication: **02.05.97**

(84) Designated contracting states:

(71) Applicant: **MITSUBISHI ELECTRIC COF**(72) Inventor: **ISHIDA HITOSHI
TOKUNAGA YUICHI**

(74) Representative:

(54) BUS CONTROLLER

(57) Abstract:

PROBLEM TO BE SOLVED: To dynamically change a retry interval and to avoid the useless retry interval even when the competition with the processing operation of an other module is caused on a bus by executing a retry operation while, changing the retry interval.

SOLUTION: A bus control system is composed of a bus control circuit 102 and a mediation circuit 104 determining the priority for the bus use request from the bus control circuit 102. The bus control circuit 102 is provided with a protocol control circuit 201 performing the data transfer processing in accordance with a bus protocol, a retry interval register 205, a retry execution frequency counter 203 and a retry interval threshold register 204, etc. When the values of the retry interval register 205 and the retry execution frequency counter 203 are compared and the values are equal when each retry processing is normally terminated, the time

(11)特許出願公開番号

特開平9-114750

(43)公開日 平成9年(1997)5月2日

(51) Int.Cl.⁸

G O 6 F 13/00

識別記号

3 0 1

片内整理番号

FI

G 0 6 F 13/00

技術表示箇所

3 0 1 Q

審査請求 未請求 請求項の数8 OL (全 25 頁)

(21)出願番号 特願平7-265586

(22)出願日 平成7年(1995)10月13日

(71)出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72)発明者 石田 仁志

東京都千代田区丸の内二丁目2番3号 三菱電機株式会社内

(72)発明者 徳永 雄一

東京都千代田区丸の内二丁目2番3号 三菱電機株式会社内

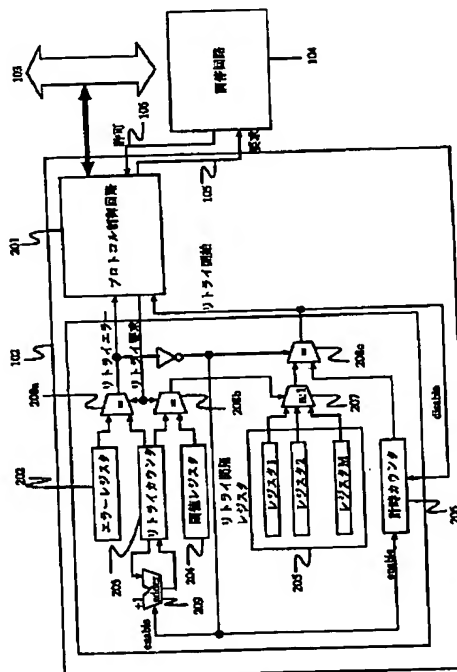
(74) 代理人 弁理士 宮田 金雄 (外3名)

(54) 【発明の名称】 バス制御装置

(57) 【要約】

【課題】 システムバス上で異常終了した処理のリトライ動作を動的に制御する、又は優先して実行することができるバス制御装置を得ることを目的とする。

【解決手段】 各モジュールはバス制御回路を備え、閾値レジスタに示されたリトライ回数で動的にリトライ間隔を変化する。また、リトライ動作を優先して実行する。



【特許請求の範囲】

【請求項1】 バスプロトコルに従ってデータ転送処理を行うプロトコル制御部とバス制御部から構成されバスに接続されたバス制御装置と、前記バス制御装置からのバス使用要求に対して優先度を決定する調停回路からなるバス制御システムにおいて、

前記バス制御装置は、

リトライ間隔を保持する1つまたは複数からなるリトライ間隔レジスタと、

リトライ実行回数を計数するリトライ実行回数カウンタと、

前記リトライ間隔レジスタの切り換えに要するリトライ回数値を保持するリトライ間隔閾値レジスタと、

最大実行リトライ回数を保持するリトライエラーレジスタと、

リトライ開始からの経過時間を計数する計時カウンタとを備え、

各リトライ処理が異常終了した際に前記リトライ間隔閾値レジスタと前記リトライ実行回数カウンタの値を比較して等しい場合、リトライ間隔を他のリトライ間隔レジスタで規定された値に切替えることによって、リトライによる再試行開始までの時間間隔を制御するようにしたことを特徴とするバス制御装置。

【請求項2】 バスプロトコルに従ってデータ転送処理を行うプロトコル制御部とバス制御部から構成されバスに接続されたバス制御装置と、前記バス制御装置からのバス使用要求に対して優先度を決定する調停回路からなるバス制御システムにおいて、

前記バス制御装置は、

リトライ間隔を保持するリトライ間隔レジスタと、

リトライ実行回数を計数するリトライ実行回数カウンタと、

バスのアイドル時間を監視し測定結果を保持するバス監視制御回路と、

前記リトライ間隔レジスタとバス監視制御回路出力の切り換えに要するリトライ回数値を保持するリトライ間隔閾値レジスタと、

最大実行リトライ回数を保持するリトライエラーレジスタと、

リトライ開始からの経過時間を計数する計時カウンタとを備え、

各リトライ処理が異常終了した際に前記リトライ間隔閾値レジスタと前記リトライ実行回数カウンタの値を比較して等しい場合、リトライ間隔をバス監視制御回路の出力で規定された値に切替えることによって、リトライによる再試行開始までの時間間隔を制御するようにしたことを特徴とするバス制御装置。

【請求項3】 バスプロトコルに従ってデータ転送処理を行うプロトコル制御部とバス制御部から構成されバスに接続されたバス制御装置と、前記バス制御装置からの

バス使用要求に対して優先度を決定する調停回路からなるバス制御システムにおいて、

前記バス制御装置は、

リトライ間隔を保持するリトライ間隔レジスタと、

リトライ実行回数を計数するリトライ実行回数カウンタと、

バスアクセス時における優先度を記録した優先度レジスタと、

前記優先度レジスタの切り換えに要するリトライ回数値を保持するリトライ間隔閾値レジスタと、

最大実行リトライ回数を保持するリトライエラーレジスタと、

リトライ開始からの経過時間を計数する計時カウンタとを備え、

各リトライ処理が異常終了した際に前記リトライ間隔閾値レジスタと前記リトライ実行回数カウンタの値を比較して等しい場合、前記バス調停回路に対するアクセス要求優先度を他の優先度レジスタの出力で規定された値に切替えることによって、前記バス調停回路に対するアクセス要求優先度を変更するようにしたことを特徴とするバス制御装置。

【請求項4】 バスプロトコルに従ってデータ転送処理を行うプロトコル制御部とバス制御部から構成されバスに接続されたバス制御装置と、前記バス制御装置からのバス使用要求に対して優先度を決定する調停回路からなるバス制御システムにおいて、

前記バス制御装置は、

該バス制御装置に対してアクセス要求を行ったモジュールを判別するモジュール判別回路と、

前記アクセス要求のあったモジュールに対してリトライ要求で応答した場合に該モジュール情報を記憶しておくモジュール情報保持回路と、

前記リトライを応答した後に受信したアクセス要求に対し、該アクセス要求を行ったモジュールが前記モジュール情報保持回路に記憶されているモジュールと同一の場合に限り該アクセス要求を受け入れる排他受信回路とを備えたことを特徴とするバス制御装置。

【請求項5】 バスプロトコルに従ってデータ転送処理を行うプロトコル制御部とバス制御部から構成されバスに接続されたバス制御装置と、前記バス制御装置からのバス使用要求に対して優先度を決定する調停回路からなるバス制御システムにおいて、

前記バス制御装置は、

受信した複数のアクセス要求情報を蓄積する要求バッファレジスタと、

前記要求バッファレジスタに保持されているアドレス情報から転送先モジュールを判別するモジュール判別回路と、

要求バッファレジスタに蓄積されているアクセス要求の1つを実行した時に実行先モジュールからリトライ要求

が返された場合、一定時間経過してからリトライを実行させるリトライウエイト回路と、リトライウエイト期間中に、前記モジュール判別回路によって別モジュールへのアクセスと判断された要求バッファレジスタ内のアクセス要求を実行する転送制御回路を備えたことを特徴とするバス制御装置。

【請求項6】 バスプロトコルに従ってデータ転送処理を行うプロトコル制御部とバス制御部から構成されバスに接続されたバス制御装置と、前記バス制御装置からのバス使用要求に対して優先度を決定する調停回路からなるバス制御システムにおいて、

前記バス制御装置は、受信した複数のアクセス要求情報を蓄積する要求バッファレジスタと、

前記要求バッファレジスタに保持されているアドレス情報から転送先モジュールを判別するモジュール判別回路と、

前記要求バッファに保持されたアクセス要求の1つを実行した時に異常転送が検出された場合、同一モジュールへのアクセス要求を保持している前記要求バッファレジスタ内のアクセス要求を消去するバッファレジスタ制御回路を備えたことを特徴とするバス制御装置。

【請求項7】 前記モジュール判別回路は、モジュールのアドレスとは処理上独立した領域を特定する領域判別回路を備え、リトライウエイト期間中に別領域へのアクセス要求を実行することを特徴とする請求項5記載のバス制御装置。

【請求項8】 前記モジュール判別回路は、モジュールのアドレスとは処理上独立した領域を特定する領域判別回路を備え、異常転送検出時に同じ領域へのアクセス要求を消去することを特徴とする請求項6記載のバス制御装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、バス上で処理が異常終了した場合に再実行処理を行なうバス制御装置に関するものである。

【0002】

【従来の技術】図16は従来例におけるシステム構成の一例を示したもので、図において、101はプロセッサ、メモリ、内部バスやバス制御回路から構成されるモジュール、102はシステムバスと内部バスの処理の送受信制御を行なうバス制御回路、103は複数のモジュールが接続されるシステムバス、104はシステムバスの使用権を調停する調停回路、105はシステムバスの使用権を獲得する時にセットされるバス要求信号、106はシステムバスの使用権を獲得した時にセットされるバス許可信号、107はプロセッサ、メモリやバス制御回路が接続される内部バス、108はアプリケーションを実行するプロセッサ、109はプログラムやデータを

保持するメモリである。

【0003】次に動作について説明する。一例として、モジュール1 101aのプロセッサ108aがモジュール2 101bのメモリ109bにアクセスするケースを説明する。プロセッサ101aは内部バス107aの使用権を獲得すると、バス制御回路102aに対してアクセスを行なう。バス制御回路102aは、内部バス107aからのアクセスを受信し、システムバス103へのアクセスであると判断すると調停回路104にバス要求信号105aを有意にする。調停回路104はモジュールからのバス要求信号の優先度を調べ、優先度の高いモジュールのバス許可信号を有意にする。優先度が同じならば、一番早く受信したモジュールのバス許可信号を有意にする。バス制御回路102aは調停回路104からシステムバス許可信号106aが有意になると、モジュール2 101bに対してアクセスを行なう。モジュール2 101b内のバス制御回路102bは、受信可能であればモジュール1 101aからのアクセスを受信し、内部バス107bの使用権を獲得した後、メモリ109bにアクセスを行ない、処理は正常に終了する。バス制御回路102bがモジュール1 101aからのアクセスを受信できない場合にはリトライ要求を出力する。モジュール1 101aは、リトライ要求を受信すると一旦システムバス103を開放した後、一定時間後にバス要求信号105aを再度、有意にする。バス許可信号106aが有意になると、モジュール2 101bに対してアクセスを行なう。モジュール2 102aは、受信可能であればアクセス要求を受信し、処理は正常に終了する。受信できないならば、再度リトライ要求を出力する。モジュール1 101aは、このようにして、一定回数リトライ要求を受信するとリトライエラーと判断し、エラー処理を実行する。

【0004】また従来、リトライ方式を定義するバスとしてIEEE896 (Futurebus+)があり、図17はビジー状態におけるリトライ回路を示したものである。図17において、901はシステムバス103の使用権を調停回路104に要求し、許可が得られた場合にシステムバス103上で処理を実行するプロトコル制御回路、902はRETRY_COUNTERフィールドとRETRY_THRESHOLDフィールドからなるBUSY_RETRY_COUNTER_CSRである。また、903はRETRY_DELAYフィールドからなるBUSY_RETRY_DELAY_CSR、904はリトライ開始時間を計数する計数カウンタ、905は加算器、906は比較器である。なお、RETRY_COUNTERフィールドはリトライ実行回数を示し、RETRY_THRESHOLDフィールドはモジュールが実行するリトライ回数の最大値を示し、RETRY_DELAYフィールドはリトライ動作を起動するまでの時間を示す。尚、102～106は図16

で相当符号を付したものと同様であり、また、CSRとはControl and Status Registersの略で、IEEE1212で定義され、レジスタの値はシステム構成時に設定される。

【0005】次に動作について説明する。プロトコル制御回路901がシステムバス103上で処理を実行中にビジーを検出すると、リトライ要求を出力する。リトライ回路はリトライ要求を受信すると、BUSY_RETRY_COUNTER_CSR902のRETRY_COUNTERフィールドとRETRY_THRESHOLDフィールドを比較する。RETRY_COUNTERの値がRETRY_THRESHOLDの値未満であれば、RETRY_COUNTERの値をインクリメントし、計時カウンタ904をイネーブルにする。計時カウンタ904とBUSY_RETRY_DELAY_CSR903に設定された値が等しくなると、リトライ開始をプロトコル制御回路901に通知すると共に、計時カウンタ904をクリアする。RETRY_COUNTERの値がRETRY_THRESHOLDの値と等しければ、プロトコル制御回路901にビジーリトライエラーを通知する。プロトコル制御回路901は、リトライ開始を受信すると再度バス要求を出力し、ビジーリトライエラーを受信すると、エラー処理の実行に入る。

【0006】

【発明が解決しようとする課題】IEEE896(Futurebus+)規格のバス仕様では、リトライ動作の実行間隔が一意に決定されているので、Futurebus+に接続された他のモジュールが、それぞれにリトライ動作を実行した場合、一定間隔で実行されるリトライ動作の処理周期が同じタイミングとなり、リトライ動作に対するリトライ要求が相次いで出力されるという問題点があった。

【0007】また、バス制御装置が複数のバスアクセス要求を蓄積する場合、一旦リトライモードに入ると、リトライを開始するまでの時間待ちの期間、他の処理も待たされるため転送処理性能を低下させるという問題点があった。

【0008】また、エラー時にエラーを発生したバスアクセス要求のみを切り離すため、アクセスの順番を重視する処理においては、データ抜けが生じ、データの整合性がとれなくなることにより処理に異常が発生するという問題点があった。

【0009】本発明は、上記のような問題点を解決するためになされたもので、バスに接続された各モジュールのリトライ動作の周期が同期した場合においても、動的にリトライ要求間隔を変化させることでリトライエラーを回避するようにしたものである。また、複数のバスアクセス要求を蓄積することによって、例えば、リトライ状態が発生してもリトライ時間待ちの期間に別のモジュールに対するバスアクセスサービスを実行することによ

り、処理効率のよいバス制御装置を提供することを目的としたものである。さらに、エラー発生時において当該エラーに関連したバスアクセス要求を消去することにより、アクセスシーケンスが重視される処理実行においても、データの整合を維持することのできるバス制御装置を提供することを目的としたものである。

【0010】

【課題を解決するための手段】第1の発明に係わるバス制御装置は、バスプロトコルに従ってデータ転送処理を行うプロトコル制御部とバス制御部から構成されバスに接続されたバス制御装置と、バス制御装置からのバス使用要求に対して優先度を決定する調停回路からなるバス制御システムにおいて、リトライ間隔を保持する1つまたは複数からなるリトライ間隔レジスタと、リトライ実行回数を計数するリトライ実行回数カウンタと、リトライ間隔レジスタの切り換えに要するリトライ回数値を保持するリトライ間隔閾値レジスタと、最大実行リトライ回数を保持するリトライエラーレジスタと、リトライ開始からの経過時間を計数する計時カウンタとを備えることにより、各リトライ処理が異常終了した際に前記リトライ間隔閾値レジスタとリトライ実行回数カウンタの値を比較して等しい場合、リトライ間隔を他のリトライ間隔レジスタで規定された値に切替えることによって、リトライによる再試行開始までの時間間隔を制御するようにしたものである。

【0011】第2の発明に係わるバス制御装置は、バスプロトコルに従ってデータ転送処理を行うプロトコル制御部とバス制御部から構成されバスに接続されたバス制御装置と、バス制御装置からのバス使用要求に対して優先度を決定する調停回路からなるバス制御システムにおいて、リトライ間隔を保持するリトライ間隔レジスタと、リトライ実行回数を計数するリトライ実行回数カウンタと、バスのアイドル時間を監視し測定結果を保持するバス監視制御回路と、リトライ間隔レジスタとバス監視制御回路出力の切り換えに要するリトライ回数値を保持するリトライ間隔閾値レジスタと、最大実行リトライ回数を保持するリトライエラーレジスタと、リトライ開始からの経過時間を計数する計時カウンタとを備えることにより、各リトライ処理が異常終了した際にリトライ間隔閾値レジスタとリトライ実行回数カウンタの値を比較して等しい場合、リトライ間隔をバス監視制御回路の出力で規定された値に切替えることによって、リトライによる再試行開始までの時間間隔を制御するようにしたものである。

【0012】第3の発明に係わるバス制御装置は、バスプロトコルに従ってデータ転送処理を行うプロトコル制御部とバス制御部から構成されバスに接続されたバス制御装置と、バス制御装置からのバス使用要求に対して優先度を決定する調停回路からなるバス制御システムにおいて、リトライ間隔を保持するリトライ間隔レジスタ

と、リトライ実行回数を計数するリトライ実行回数カウンタと、バスアクセス時における優先度を記録した優先度レジスタと、優先度レジスタの切り換えに要するリトライ回数値を保持するリトライ間隔閾値レジスタと、最大実行リトライ回数を保持するリトライエラーレジスタと、リトライ開始からの経過時間を計数する計時カウンタとを備えることにより、各リトライ処理が異常終了した際にリトライ間隔閾値レジスタとリトライ実行回数カウンタの値を比較して等しい場合、バス調停回路に対するアクセス要求優先度を他の優先度レジスタの出力で規定された値に切替えることによって、バス調停回路に対するアクセス要求優先度を変更するようにしたものである。

【0013】第4の発明に係わるバス制御装置は、バスプロトコルに従ってデータ転送処理を行うプロトコル制御部とバス制御部から構成されバスに接続されたバス制御装置と、バス制御装置からのバス使用要求に対して優先度を決定する調停回路からなるバス制御システムにおいて、バス制御装置に対してアクセス要求を行ったモジュールを判別するモジュール判別回路と、アクセス要求のあったモジュールに対してリトライ要求で応答した場合に該モジュール情報を記憶しておくモジュール情報保持回路と、リトライを応答した後に受信したアクセス要求に対し、該アクセス要求を行ったモジュールがモジュール保持回路に記憶されているモジュールと同一の場合に限り該アクセス要求を受け入れる排他受信回路とを備えるようにしたものである。

【0014】第5の発明に係わるバス制御装置は、バスプロトコルに従ってデータ転送処理を行うプロトコル制御部とバス制御部から構成されバスに接続されたバス制御装置と、バス制御装置からのバス使用要求に対して優先度を決定する調停回路からなるバス制御システムにおいて、受信した複数のアクセス要求情報を蓄積する要求バッファレジスタと、要求バッファレジスタに保持されているアドレス情報から転送先モジュールを判別するモジュール判別回路と、要求バッファレジスタに蓄積されているアクセス要求の1つを実行した時に実行先モジュールからリトライ要求が返された場合、一定時間経過してからリトライを実行させるリトライウエイト回路と、リトライウエイト期間中に、上記モジュール判別回路によって別モジュールへのアクセスと判断された要求バッファレジスタ内のアクセス要求を実行する転送制御回路とを備えるようにしたものである。

【0015】第6の発明に係わるバス制御装置は、バスプロトコルに従ってデータ転送処理を行うプロトコル制御部とバス制御部から構成されバスに接続されたバス制御装置と、バス制御装置からのバス使用要求に対して優先度を決定する調停回路からなるバス制御システムにおいて、受信した複数のアクセス要求情報を蓄積する要求バッファレジスタと、要求バッファレジスタに保持され

ているアドレス情報から転送先モジュールを判別するモジュール判別回路と、要求バッファに保持されたアクセス要求の1つを実行した時に異常転送が検出された場合、同一モジュールへのアクセス要求を保持している要求バッファレジスタ内のアクセス要求を消去するバッファレジスタ制御回路を備えるようにしたものである。

【0016】第7の発明は第5の発明におけるバス制御装置において、モジュール判別回路に、モジュールのアドレスとは処理上独立した領域を特定する領域判別回路を備えるようにし、リトライウエイト期間中に別領域へのアクセス要求を実行するようにしたものである。

【0017】第8の発明は第6の発明におけるバス制御装置において、モジュール判別回路に、モジュールのアドレスとは処理上独立した領域を特定する領域判別回路を備えるようにし、異常転送検出時に同じ領域へのアクセス要求をキャンセルするようにしたものである。

【0018】

【発明の実施の形態】

実施の形態1. 本発明の第1の実施の形態について、図1、図2に基づいて説明する。図1はバス制御回路の構成図であり、図中、201はシステムバスのプロトコル制御を行うプロトコル制御回路、202はリトライエラーを検知するためのリトライ回数を保持しておくエラーレジスタ、203はリトライ発生時における現在までのリトライ回数を保持するリトライカウンタ、204はリトライ間隔を切替えるまでに実行するリトライの回数を保持しておく閾値レジスタである。また、205は各々異なったリトライ動作の実行間隔時間を保持する複数からなるリトライ間隔レジスタ群、206はリトライ要求を受信してからリトライ動作を起動するまでの時間を計測する計時カウンタ、207は複数のリトライ間隔レジスタ群から一個のレジスタを選択するセクタ、208cはセクタ207の出力結果と計時カウンタ206の出力を比較する比較器、209はリトライ実行の都度、リトライカウンタ内容をインCREMENTしていく加算器である。なお、図において、符号102～106は図16の相当符号と同様の要素を示す。

【0019】次に動作について、図1、および図2を用いて説明する。なお、図2では、予め、エラーレジスタ202に値“4”を、閾値レジスタ204に値“2”を、更にリトライ間隔レジスタ1に値“ T_1 ”を、リトライ間隔レジスタ2に値“ T_2 ”を設定しておいた場合において、リトライエラーが発生した時の様子を表している。プロトコル制御回路201は、システムバス103に対する転送要求を受信すると、バス要求信号105を有意にする(時刻t1)。調停回路104からのバス許可信号106が有意になった時点で(時刻t2)、システムバス103にデータ転送処理を起動する(時刻t3)。ここで、処理が正常に終了した時は、リトライカウンタ203、計時カウンタ206はリセットされ、プ

ロトコル制御回路201は次のシステムバス要求が来るまでアイドル状態となる。

【0020】一方、システムバス103からリトライ要求信号を受信した場合(時刻 t_4)、プロトコル制御回路201はデータ転送処理を終了させ(時刻 t_5)、リトライ制御回路に対してリトライ要求を出力する。プロトコル制御回路201からリトライ要求を受信すると、比較器208a、208bは各々リトライカウンタ203とエラーレジスタ202、およびリトライカウンタ203と閾値レジスタ204の内容を比較する。

(ケース1)リトライカウンタ203の値が閾値レジスタ204、及びエラーレジスタ201のいずれの値とも等しくない場合(時刻 t_5 、 t_7 、 t_9)には、加算器209がイネーブルになり、リトライカウンタ203の値を1加算する。それと共に、計時カウンタ206がイネーブルになり、時間を計り始める。計時カウンタ206は、リトライ開始によりディスエーブルになるまで時間を計り続ける。

(ケース2)リトライカウンタ203の値と閾値レジスタ204の値("2")が等しい場合(時刻 t_8)には、リトライ間隔を切替えるために、セクタ207にセレクト信号が出力される。この例では、次のリトライまでの時間間隔がこれまでの" T_1 "から" T_2 "に変化している様子がわかる。

(ケース3)リトライカウンタ203の値とエラーレジスタ202の値("4")が等しい場合(時刻 t_{10})には、プロトコル制御回路201にリトライエラーを通知すると共に、リトライ開始信号をディスイネーブルにし、リトライカウンタ203および計時カウンタ206をリセットして一連のリトライ動作を終了する。

【0021】比較器208cは、リトライエラーになるまで、現在選択されているリトライ間隔レジスタ205が保持している値と計時カウンタ206の値を比較していて、等しくなった場合に、プロトコル制御回路201にリトライ開始信号を出力すると共に計時カウンタをリセットする。プロトコル制御回路201はリトライ開始信号を受信すると(時刻 t_6)、調停回路104にバス要求信号105を出力し、調停回路104からバス使用权が得られると、システムバス103に対して処理の再実行を試みる。

【0022】このように、第1の実施形態によれば、リトライ間隔を変更しながらリトライ動作を実行するようにしたので、システムバス103上で他のモジュールによる処理動作と例え、動作が同期した場合においても、リトライ間隔を動的に変化させることができるので、無用なリトライ動作を回避することができるという効果がある。

【0023】実施の形態2. この発明の第2の実施の形態について図3、図4に基づいて説明する。図3はバス制御回路102の中のバス要求部に関する構成図であ

り、図において、302はシステムバス103上でのアイドル時間を計測する監視回路、301は監視回路の出力を保持する監視レジスタである。尚、図において、符号102乃至106および201~209は各々図1、図2記載の相当符号と同様のものである。

【0024】次に動作について説明する。なお、予め、図3のエラーレジスタ202には値"4"を、閾値レジスタ204には値"2"を、またリトライ間隔レジスタ1には値" T_1 "を設定し、リトライ動作の結果、処理が正常に終了した場合を表している。プロトコル制御回路201は、システムバスへの要求を受信するとバス要求信号105を有意にする(時刻 t_1)。調停回路104からのバス許可信号106が有意になると(時刻 t_2)、システムバス103に対してデータ転送処理を起動する(時刻 t_3)。データ転送処理が正常に終了すると、リトライカウンタ203、計時カウンタ206はリセットされ、プロトコル制御回路201は次のシステムバス要求が来るまでアイドル状態となる。ここで、システムバス103からリトライ要求信号を受信した場合(時刻 t_4)には、プロトコル制御回路201は処理を終了させ(時刻 t_5)、リトライ制御回路に対してリトライ要求を出力する。

【0025】プロトコル制御回路201からリトライ要求を受信すると、比較器208a、208bはリトライカウンタ203とエラーレジスタ202、およびリトライカウンタ203と閾値レジスタ204の値を比較する。

(ケース1)リトライカウンタ203の値が閾値レジスタ204、エラーレジスタ202の両方の値と等しくない場合(時刻 t_5 、 t_8)には、加算器209がイネーブルになり、リトライカウンタ203の値を1加算する。それと共に、計時カウンタ206がイネーブルになり、時間を計り始める。計時カウンタ206はリトライ開始によりディスエーブルになるまで時間を計り続ける。

(ケース2)リトライカウンタ203の値と閾値レジスタ204の値が等しい場合(時刻 t_{10})には、リトライ間隔を監視レジスタ301の値に切替えるために、セクタ207にセレクト信号を出力する。この例では、次のリトライまでの時間間隔がこれまでの" T_1 "からシステムバス上の監視レジスタの値" T_2 "に変化している様子がわかる。

(ケース3)リトライカウンタ203の値とエラーレジスタ201の値が等しい場合は、プロトコル制御回路201にリトライエラーを通知すると共に、リトライ開始信号をディスイネーブルにし、リトライカウンタ203と計時カウンタ206をリセットして一連のリトライ処理を終了する。

【0026】比較器208cは、リトライエラーになるまで、セクタ207の出力結果と計時カウンタ206

の値を比較する。一致すると、プロトコル制御回路201にリトライ開始信号を出力すると共に、計時カウンタをリセットしディスエーブルにする。監視回路302は、プロトコル制御回路201からリトライ要求を受信すると(時刻 t_5)、システムバス103上の処理を監視し、システムバス103上で処理が行われていない時間(この例では T_2)を測定し、監視レジスタ301に測定結果を出力する。プロトコル制御回路201は、リトライ開始信号を受信すると、調停回路104に対してバス要求信号105を有意にし、調停回路104からのバス許可信号106が有意になると、システムバスに対してデータ転送処理を再実行する。

【0027】このように、本実施形態によると、システムバス上の他の周期処理とリトライ動作が同期した場合、システムバス上の現在の負荷状況に従ってリトライ間隔を動的に変化させながらリトライ動作を実行するようにしたので、システムバス上の他のモジュールによる動作周期との重なりを極力回避することができ、無用なリトライ動作を回避し、少ないリトライ回数でデータ転送を終了することができるという効果がある。

【0028】実施の形態3。この発明の第3の実施の形態について、図5、図6に基づいて説明する。図5はバス制御回路102の中のバス要求部に関する構成図であり、図において、401はシステムバス要求信号105の優先度を指定する優先度レジスタである。なお、符号102乃至106、および201~209は各々図1、図2の相当符号と同一要素を表わす。

【0029】次に動作について説明する。なお、図5では予め、エラーレジスタ202に値"4"を、閾値レジスタ204に値"2"を、またリトライ間隔レジスタ205に値" T_1 "を設定し、リトライ動作において処理が正常に終了した場合を表している。プロトコル制御回路201は、システムバスへの要求を受信するとバス要求信号105を有意にする(時刻 t_1)。調停回路104からのバス許可信号106が有意になると(時刻 t_2)、システムバス103に対してデータ転送処理を起動し(時刻 t_3)、処理が正常に終了するとリトライカウンタ203、計時カウンタ206はリセットされ、プロトコル制御回路201は次のシステムバス要求が来るまでアイドル状態となる。ここで、システムバス103からリトライ要求信号を受信した場合(時刻 t_4)には、プロトコル制御回路201はデータ転送処理を終了させ(時刻 t_5)リトライ制御回路に対してリトライ要求を出力する。

【0030】プロトコル制御回路201からリトライ要求を受信すると、比較器208a、208bは、各々リトライカウンタ203とエラーレジスタ202、およびリトライカウンタ203と閾値レジスタ204の値を比較する。

(ケース1)リトライカウンタ203の値が閾値レジ

スタ204、エラーレジスタ202のいずれの値とも等しくない場合(時刻 t_5)には、加算器209がイネーブルになり、リトライカウンタ203の値を1加算する。それと共に、計時カウンタ206がイネーブルになり、時間計測を開始し、リトライ開始信号が出力されてディスエーブルになるまで時間を計り続ける。

(ケース2)リトライカウンタ203の値と閾値レジスタ204の値が等しい場合(時刻 t_{10})には、システムバス要求信号105の優先度を切替えるために、プロトコル制御回路201に切替信号を出力する。

(ケース3)リトライカウンタ203の値とエラーレジスタ202の値が等しい場合、プロトコル制御回路201にリトライエラーを通知すると共に、リトライ開始信号をディスイネーブルにする。

【0031】比較器208cは、リトライエラーになるまでセレクト207の値と計時カウンタ206の値を比較し、等しい場合には、プロトコル制御回路201にリトライ開始信号を出力すると共に計時カウンタをリセットしディスエーブルにする。プロトコル制御回路201は、リトライ開始信号と切替信号の両方を受信すると、高い優先度のバス要求信号105を有意にする(時刻 t_{10})。調停回路104からバス使用権が得られると、システムバスに対してデータ転送処理を再実行する。

【0032】本実施形態によれば、システムバス上の他のモジュールの処理周期と同期した場合においても、一定回数リトライ動作を実行した後は、バス要求信号の優先度を高くしてバス使用要求を出力するようにしたので、優先処理によって徒らにリトライ要求回数を増やすことなく処理を行うことができるという効果がある。

【0033】実施の形態4。本発明の第4の実施の形態について、図7、図8に基づいて説明する。図7は、バス制御装置を示す構成図であり、501はマスタモジュールからシステムバス経由で転送されるデータを受信し内部バスへ転送するスレーブモジュール、502a、502bは各々マスタモジュール1および2、503はスレーブモジュール501内のバス制御装置、504はスレーブモジュール501の内部バス、505はスレーブモジュール501がシステムバス103から受信したデータを内部バス504へ転送するためのデータバッファ、506はバス調停回路104が出力する許可信号106a、106bを入力して、バスを使用しているマスタモジュール番号を識別するエンコード、507はリトライ要求発生時にバスを使用しているマスタモジュールの番号を記憶するラッチである。また、508はラッチ507の保持しているマスタモジュールの番号と現在システムバスを使用しているマスタモジュールの番号を比較する比較器、509はデータバッファ505内部にデータが転送されずに残っている場合にセットされるバッファエンパティ信号、510はバッファエンパティ信号509がリセットまたは比較器508の出力がリセットされ

ている時にバスアクセス要求があった場合にシステムバスへリトライ要求を発生するリトライ要求信号である。511はバスアクセスの開始を示すアクセス信号線である。

【0034】次に動作について説明する。尚、図8はマスタモジュール2(502b)のアクセスの直後にマスタモジュール1(502a)がアクセスし、リトライ要求が成功する例を示したものである。マスタモジュール1 502a、およびマスタモジュール2 502bは、スレーブモジュール501へのデータ転送要求が発生すると、バス調停回路104へバス要求信号105a、105bをセットしてバス使用要求を出し、バス調停回路104がセットするバス使用許可信号106a、106bを受けて、システムバス103へバスプロトコルに従ってデータを出力する。システムバス103の使用者は1モジュールに限られるので、バス許可信号106aと106bが、同時にセットされることはない。

【0035】マスタモジュール1 502aがシステムバス103を獲得してスレーブモジュール501とのデータ転送が開始されると、スレーブモジュール501のバス制御回路503は一旦データをデータバッファ505へ格納し、マスターモジュールとのバス接続を開放した後、データバッファ505から内部バス504へのデータ転送を開始する。もし、それ以前に受信したデータがまだ内部バス504に全て転送され終えてなく、データバッファ505にデータが残っている場合、即ちバッファエンパティ信号509がリセットされている時、リトライ要求信号510がセットされマスタモジュール1 501aへリトライ要求として通知される。

【0036】一方、リトライ要求信号510のセットによって、ラッチ507が入力保持になり、エンコーダ506の出力、即ち、この場合はリトライ要求を発生した時のマスタモジュール1 502aの番号"1"がラッチ507に保持される。

【0037】このような状態で、マスタモジュール2 502bがスレーブモジュール501にデータを転送した時、スレーブモジュール501のバス制御回路503は比較器508において、ラッチ507出力とエンコーダ506出力を比較する。ラッチ507出力はマスタモジュール1 502aを示す番号"1"で、エンコーダ506出力はマスタモジュール2 502bを示す番号"2"なので、比較結果は為("0")となるので、リトライ要求信号510がセットされて、マスタモジュール2 502bへリトライ要求が出力される。

【0038】次に、マスタモジュール1 502aがスレーブモジュール501にリトライアクセスを要求した時、ラッチ507出力とエンコーダ506出力は同じマスタモジュール1 502aを示すので、比較器508は真("1")を示し、この時データバッファ505のデータが既に内部バス504へデータ転送を終了し、バ

ッファエンパティ信号509がセットされていれば、リトライ要求信号をリセットし、アクセス要求を受け入れる。このようにして、ラッチ507に記憶されたマスタモジュール1 502aからのリトライアクセスが受信されるまでは、他のアクセス要求を受け付けないため、リトライアクセスが再びリトライ要求でリジェクトされることはなく、確実に受信できる。

【0039】また、本実施の形態ではマスタ情報保持回路としてラッチ507を用い、1つのマスタ情報を保持しているが、複数のマスタ情報を保持し、アクセス要求の受け入れに順番付けをしても良い。

【0040】実施の形態5. 本発明の第5の実施形態について、図9乃至図11に基づいて説明する。図9は、バス制御回路中のマスタモジュールに関する構成図であり、マスタモジュールのバス制御部は、内部バス上位から転送されるアドレスとデータを受信し、それをシステムバスを経由してアドレスに対応するスレーブモジュールへ転送するものである。図において、501a、501bは各々スレーブモジュール1および2、503はマスタモジュールであり、103はマスターモジュールとスレーブモジュールを接続するシステムバスである。600はマスタモジュール503内のバス制御部、610は内部バス、601a~601cは内部バス610から受信したアドレスおよびデータをシステムバスへ転送するまでの期間保持しておくための要求バッファレジスタ、602a~602cは要求バッファレジスタ601a~601cに格納されているアドレスから個々のスレーブモジュールを判別するID番号に変換するための変換マップ、603はスレーブモジュールからのリトライ要求を受信した時にリトライ開始までの時間を管理するリトライウエイトタイマ、604は要求バッファレジスタ601a~601cの内のどの要求をシステムバスへ転送するかを制御するための転送制御回路である。また、605は要求バッファレジスタ601a~601cとシステムバスとのバスを切替えるためのセレクト、606a~606cは要求バッファレジスタ601a~601cから出力されるアドレス信号、607はスレーブモジュールからのリトライ要求信号、608はリトライウエイトタイマ603からの出力であるリトライウエイト信号、609はセレクト605のセレクト出力を指定するセレクト信号、611a~611cは変換マップ602a~602cの出力であるスレーブモジュールIDである。

【0041】図10は、転送制御回路604の動作フローを示した図である。また、図11は、変換マップ602a~602cの詳細を示した図である。

【0042】次に動作について説明する。マスタモジュール503では、内部バス610から受信しシステムバスへ出力するアドレスとデータを、要求バッファレジスタ601a~601cの空いているレジスタに格納す

る。内部バス610はアドレス、およびデータを転送した後に開放され、次のバスサイクルに移ることができる。この様にして、内部バス610からシステムバス103へのアクセス要求は、システムバス103の使用権獲得とは独立に次々に受信され、複数の要求バッファレジスタへ分配されて格納される。

【0043】要求バッファレジスタ601a~601cに格納されたアドレス信号606a~606cは、各々アドレス変換マップ602a~602cによって送信先のスレーブモジュールを特定するID値へ変換され、転送制御回路604へ出力される。転送制御回路604はセレクト605へセレクト信号609を送り、要求バッファレジスタ601a~601cからのアクセス要求を順番にシステムバス103へ転送する。

【0044】ここで、要求バッファレジスタ601a~601cに、各々スレーブモジュール1501a、スレーブモジュール1501a、スレーブモジュール2501bへのアクセス要求が格納されていたと仮定する。要求バッファレジスタ601aの要求をシステムバス103へ転送した後、スレーブモジュール1501aからリトライ要求が応答された場合、リトライウエイト回路はタイマを起動し、予め設定されたリトライ待ち時間の間待ち状態に入る。この時、リトライウエイトタイマ603は、リトライ待ち状態にあることを示すためにリトライウエイト信号608をセットする。

【0045】転送制御回路604は、リトライウエイト信号608のセットを検出すると、現在アクセス要求中のスレーブIDを保存した後、他の要求バッファレジスタ601b、および601cのアドレス変換マップ出力611b、611cとの値を比較する。その結果、要求バッファレジスタ601cの転送先が他スレーブモジュール、即ちスレーブモジュール2501bへのアクセス要求であることを検知すると、セレクト605へのセレクト信号609を制御し、要求バッファレジスタ601cに対する転送要求処理を行なう。

【0046】要求バッファレジスタ601cに対する転送要求を終了し、リトライウエイト信号608がリセットされた時点で再び要求バッファレジスタ601aに対する転送要求の実行を試み、これが成功すると、続いて要求バッファレジスタ601bの転送要求を実行することにより、全ての転送処理を終了する。このようにリトライウエイト期間を利用して、リトライ要求を返したスレーブモジュールとは別のスレーブモジュールとの転送動作を継続することにより、ウエイト期間中におけるバス制御回路の処理低下を防ぐことができる。

【0047】実施の形態6. この発明の第6の実施形態について、図12、図13に基づいて説明する。図12は、バス制御回路中のマスタモジュールに関する構成図であり、図において、701は異常転送が発生した時に、要求バッファレジスタ601a~601cをクリア

するリセット信号を発生するためのバッファレジスタ制御回路、703a~703cはこれらレジスタをクリアするためのレジスタリセット信号であり、702はスレーブモジュール501からの異常転送検出信号である。また、図13はバッファレジスタ制御回路701の内部構成を示す図である。

【0048】次に動作について説明する。内部バス610からの転送要求を要求バッファレジスタ601a~601cに格納し、内部バスとは独立に各々システムバス103へアクセス要求を転送することは、先の第5の実施の形態と同様である。ここで、要求バッファレジスタ601a~601cに各々スレーブモジュール1501a、スレーブモジュール1501a、スレーブモジュール2501bへのアクセス要求が格納されていたと仮定する。要求バッファレジスタ601aの要求をシステムバス103へ転送した時、何らかのバス転送異常が発生してスレーブモジュール1501aから異常転送検出信号702が応答された場合、マスタモジュール503はシステムバス103のバスサイクルを中止する。

【0049】バッファレジスタ制御回路701は、異常転送検出信号702がセットされると、現在バスへ転送している要求バッファレジスタ601a中のスレーブモジュール501aのマップ変換出力であるスレーブモジュールID611aと、全要求バッファレジスタ601a~601cのマップ変換出力であるスレーブモジュールID611a~611cを比較し、同じスレーブIDを持つ要求バッファレジスタ601a、601bへリセット信号703a、703bを送り、要求バッファレジスタの内容をクリアする。マスタモジュール503は、残る要求バッファレジスタ601c中に格納されているスレーブモジュール501bのマップ変換結果であるスレーブID611cを要求をスレーブモジュール2501bへ転送し、すべての転送を終了する。このように、スレーブモジュール1501aへのデータ転送で異常転送が発生した時に、以降、該モジュールが回復するまではスレーブモジュール1501aに対するデータ転送を全てキャンセルすることにより、一連の連続したデータ転送において、データ抜けによる処理動作の異常を防ぐことができる。

【0050】実施の形態7. 本発明の第7の実施形態について、図14、図15に基づいて説明する。図14は、バス制御回路中のマスタ制御部に関する構成図であり、図において、801a~801cはアドレスを処理対応に区分された領域を判別するためのID番号へ変換する変換マップ、802a~802cは変換マップ801a~801cによって出力される領域区分ID信号である。区分された領域とは、作業用メモリ領域、制御管理レジスタ領域、あるいはI/O領域などのように、計算機が処理する上で、完全に別の目的で使われるアド

レス領域を指す。図におけるその他の構成要素は実施の形態5に記載の相当番号と同一である。また図15は、変換マップ801a~801cの詳細を記載した図である。

【0051】次に、動作について説明する。ここで、要求バッファレジスタ601a~601cには、スレーブモジュール1501aへのアクセス要求として、各々メモリ領域1、メモリ領域2、I/O領域に対するアクセス要求が格納されていたとする。ここでメモリ領域とは例えば、プロセッサ演算の命令あるいはデータが格納される領域のことであり、I/O領域とはDMA制御等の制御コマンドが格納される領域であって、お互いが干渉し合うことはない。

【0052】要求バッファレジスタ601aの要求をシステムバス103へ転送し、スレーブモジュール1501aからリトライ要求が応答された場合、転送制御回路604は、各変換マップの出力である要求バッファレジスタのアドレスの領域区分IDを比較する。そして、要求バッファレジスタ601cの転送先がメモリ領域とは異なる他領域、即ちI/O領域へのアクセス要求であることを検知して、リトライウエイトタイムによる先のメモリ領域に対するリトライ開始待ちの期間を用いて、要求バッファレジスタ601cに対する転送要求を行なう。

【0053】要求バッファレジスタ601cの要求に対する転送処理が終了し、リトライウエイト信号608がリセットされた時点で、再び要求バッファレジスタ601aの要求、即ちメモリ領域に対する転送を再開し、続いて要求バッファレジスタ601bに対する転送要求（これも同じくメモリに対する転送）を実行して、全ての転送処理を終了する。特に、領域区分が異なるアドレスへのアクセスは、順序が入れ替わっても問題がないため、リトライウエイト時間を利用して先に転送することができる。

【0054】また、図12において、アドレスからスレーブIDへの変換マップ602a~602cをアドレスから領域区分IDへの変換マップ801a~801cに置き換え、要求バッファレジスタ601a~601cに対しては、メモリ領域、メモリ領域、I/O領域へのアクセス要求が格納されていた場合、要求バッファレジスタ601aの要求に対しスレーブモジュール1501aから異常転送検出が応答された時、バッファレジスタ制御回路701は、各要求バッファレジスタのアドレスの領域区分IDを比較し、転送中の要求バッファレジスタ601aの領域区分IDと要求バッファレジスタ601bの領域区分IDが同じ、即ちメモリ領域へのアクセス要求であることを検知して要求バッファレジスタ601a、601bをリセットし、マスタモジュール503は残る要求バッファレジスタ601cの要求をI/O領域へ転送し、全ての転送を終える。このように、領域区

分が異なるアドレスへのアクセスは、別領域へのデータが消去されても処理上問題がないため、異常転送検出時は同じ領域区分のデータのみを消去するだけで良い。

【0055】

【発明の効果】本発明は、以上説明したようにして構成されているので、以下に記載されるような効果を奏する。

【0056】この発明によれば、リトライ間隔を変更しながらリトライ動作を実行するようにしたので、バス上において他のモジュールの処理動作と競合を起こした場合においても、リトライ間隔を動的に変化させることができるので、無用なリトライ動作を回避することができるという効果がある。

【0057】また、この発明によれば、バスの負荷状況に応じてリトライ間隔を動的に変化させるようにしたので、バス上の他のモジュールと処理周期が同期した場合においても、動作周期の重なりを極力回避することができるので、効率の良いリトライ動作を実行することができる。

【0058】また、この発明によれば、一定回数のリトライ動作を実行した後は、バス使用要求に対する優先度を高くするようにしたので、バス上の他のモジュール動作と競合した場合においても、これを回避して効率のよいリトライ処理を行うことができる。

【0059】また、この発明によれば、アクセス要求のあったモジュール情報を保持し、リトライ状態発生後は、保持回路に記録していると同じモジュールからのアクセス要求に限って受理するようにしたので、他のモジュールの処理要求周期との同期によるリトライエラーを回避することができる。

【0060】また、この発明によれば、リトライウエイト期間を利用して、リトライ要求を返したモジュールとは別のモジュールに対して転送動作を継続するようにしたので、リトライ時間待ち期間中におけるバス制御回路の処理低下を防ぐことができる。

【0061】また、この発明によれば、データ転送で異常転送が発生した際に、以降、該モジュールが回復するまでこのモジュールに対するデータ転送を全てキャンセルするようにしたので、一連の連続したデータ転送においてデータ抜けによる処理動作の異常を回避することができる。

【0062】さらに、この発明によれば、各モジュールを処理上独立した領域区分に分割してアクセスするようにしたので、異なる領域に対するアクセス要求実行が可能となり、リトライ時間待ちによる転送処理性能低下および異常転送のデータ抜けによる処理の異常を回避することができる。

【図面の簡単な説明】

【図1】 この発明の第1の実施の形態を示す構成図である。

【図2】 この発明の第1の実施の形態におけるバス制御装置のタイミングチャートを示す図である。

【図3】 この発明の第2の実施の形態を示す構成図である。

【図4】 この発明の第2の実施の形態におけるバス制御装置のタイミングチャートである。

【図5】 この発明の第3の実施の形態を示す構成図である。

【図6】 この発明の第2の実施の形態におけるバス制御装置のタイミングチャートである。

【図7】 この発明の第4の実施の形態を示す構成図である。

【図8】 この発明の第4の実施の形態におけるバス制御装置のタイミングチャートである。

【図9】 この発明の第5の実施の形態を示す構成図である。

【図10】 この発明の第5の実施の形態における転送制御回路の動作を示すフローチャートである。

【図11】 この発明の第5の実施の形態における変換マップの構成図である。

【図12】 この発明の第6の実施の形態を示す構成図である。

【図13】 この発明の第6の実施の形態におけるバッファレジスタ制御回路を示す構成図である。

【図14】 この発明の第7の実施の形態を示す構成図である。

【図15】 この発明の第7の実施の形態における変換マップの構成図である。

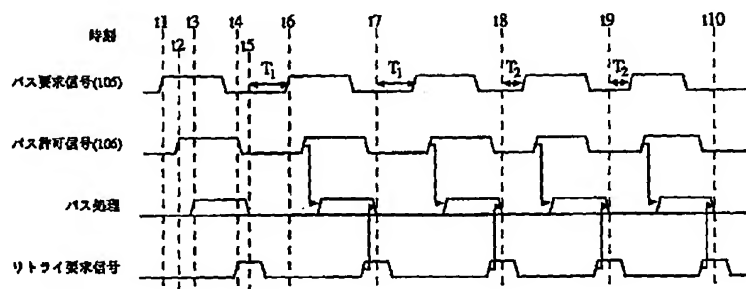
【図16】 従来例におけるバス制御装置を示す構成図である。

【図17】 従来例におけるバス制御装置を示す構成図である。

【符号の説明】

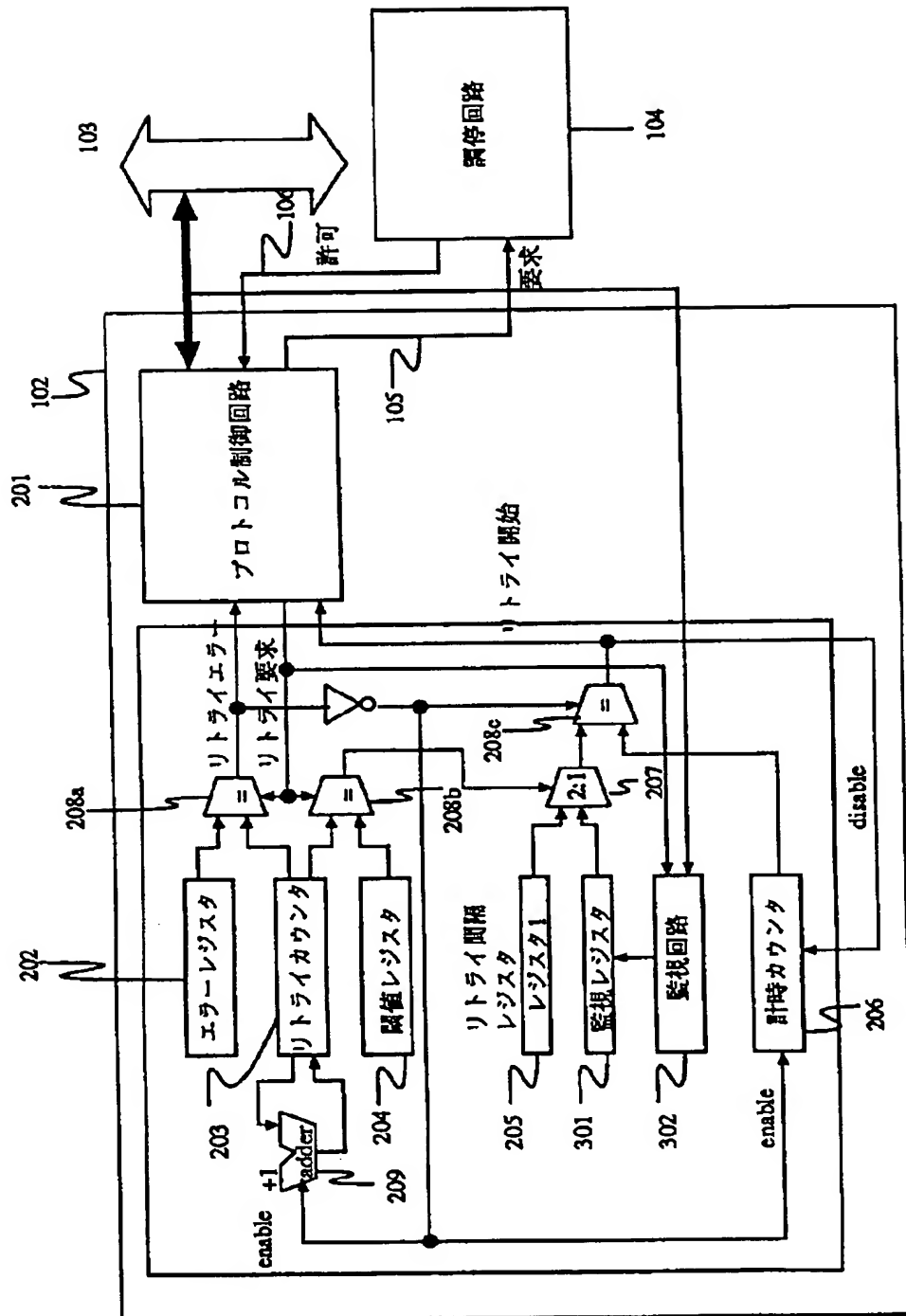
101 モジュール、102 バス制御回路、103 システムバス、104 調停回路、105 バス要求信号、106 バス許可信号、107 内部バス、108 プロセッサ、109 メモリ、201 プロトコル制御回路、202 エラーレジスタ、203 リトライカウンタ、204 閾値レジスタ、205 リトライ間隔レジスタ、206 計時カウンタ、207 セレクタ、208 比較器、209 加算器、301 監視レジスタ、302 監視回路、401 優先度レジスタ、501 スレーブモジュール、502 マスタモジュール、503 バス制御装置、504 内部バス、505 データバッファ、506 エンコーダ、507 ラッチ、508 比較器、600 バス制御部、601 要求バッファレジスタ、602 アドレス→スレーブID変換マップ、603 リトライウエイトタイマ、604 転送制御回路、605 セレクタ、609 セレクト信号、610 ローカルバス、701 バッファレジスタ制御回路、703 レジスタリセット信号、702 異常転送検出信号、801 アドレス→領域区分ID変換マップ。

【図2】

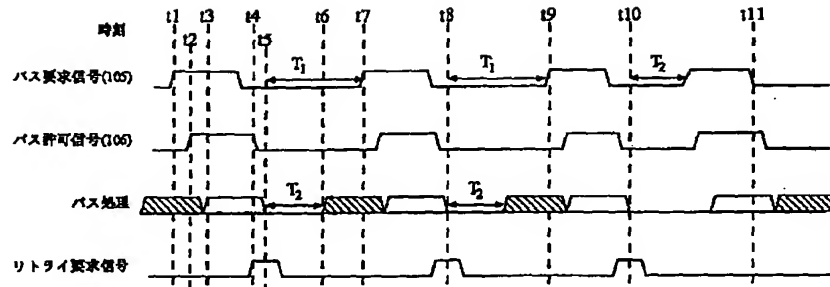


[illegible]

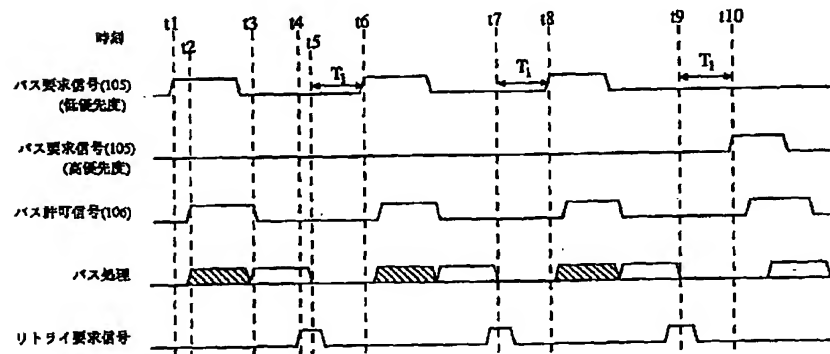
【図3】



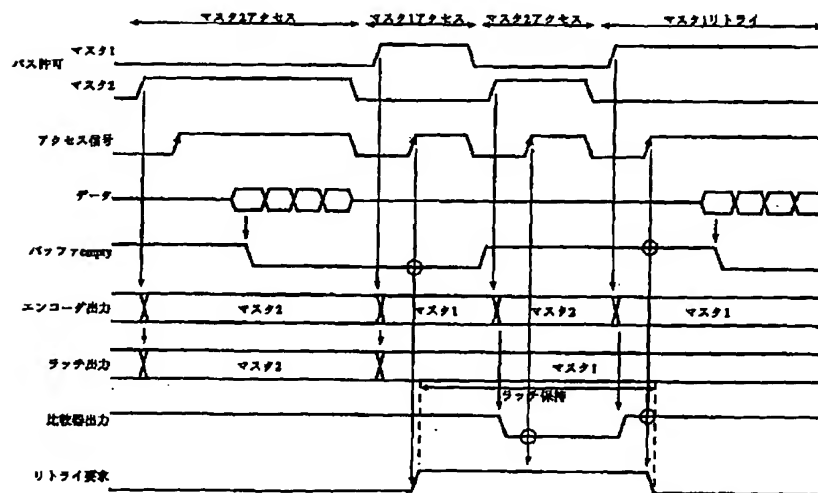
【図4】



【図6】

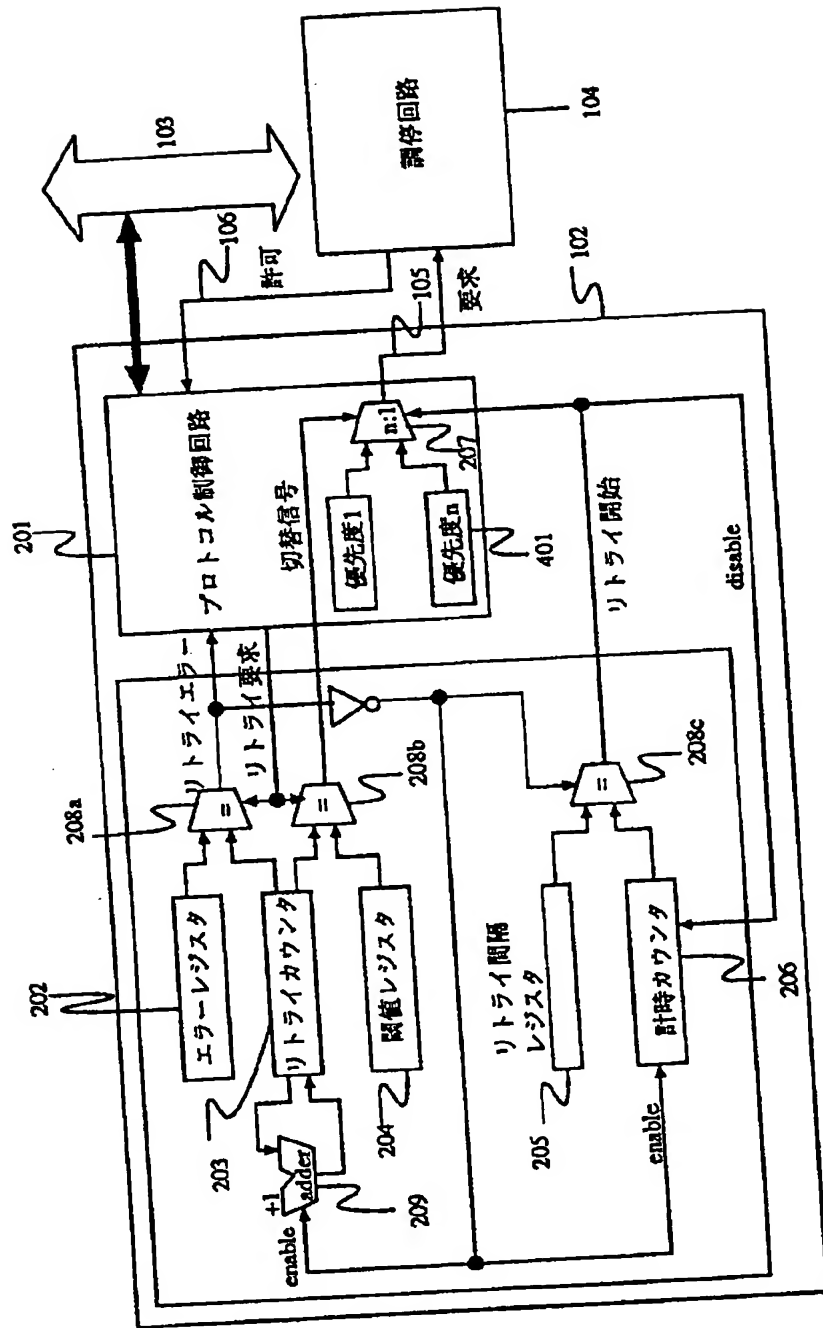


【図8】

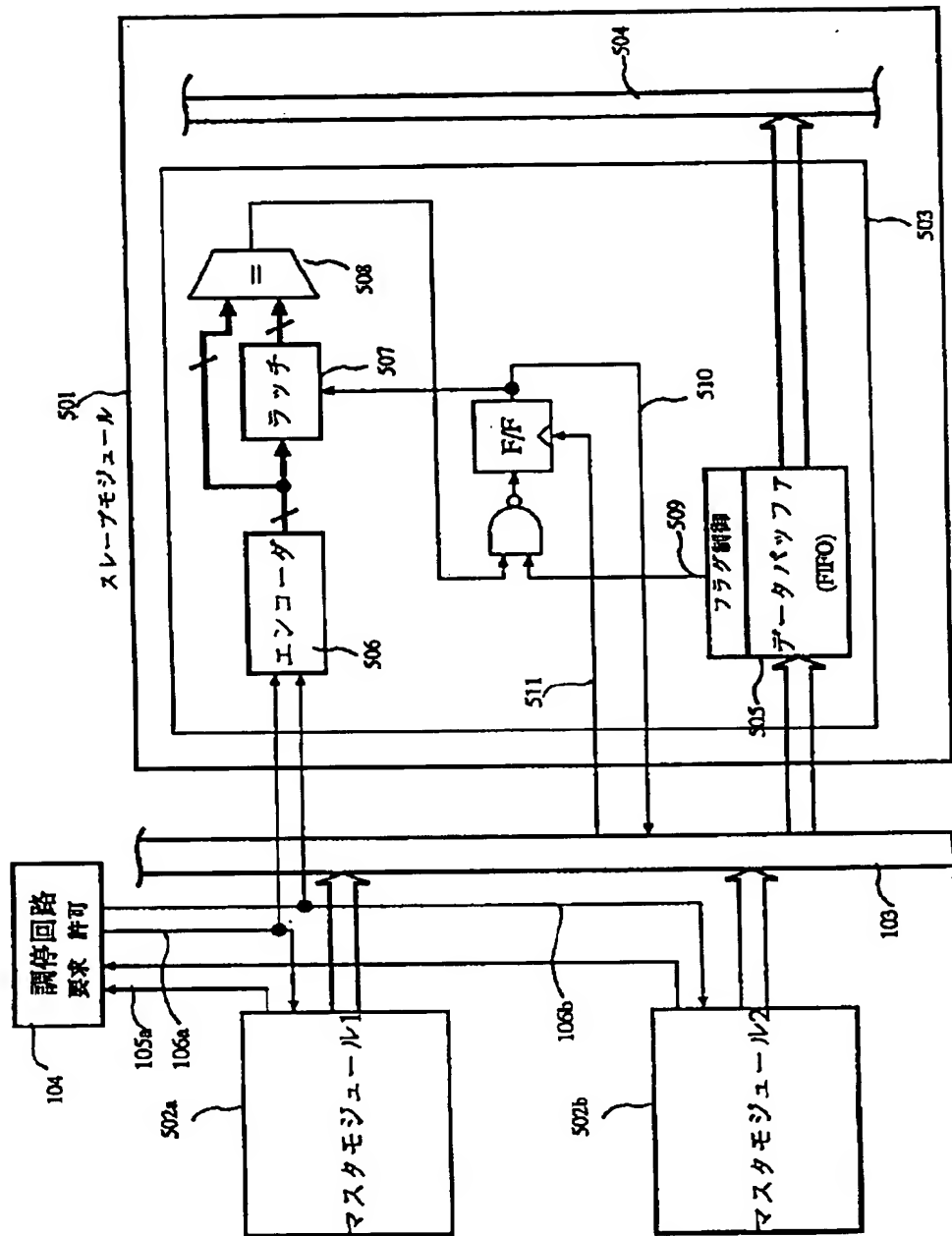


(15)

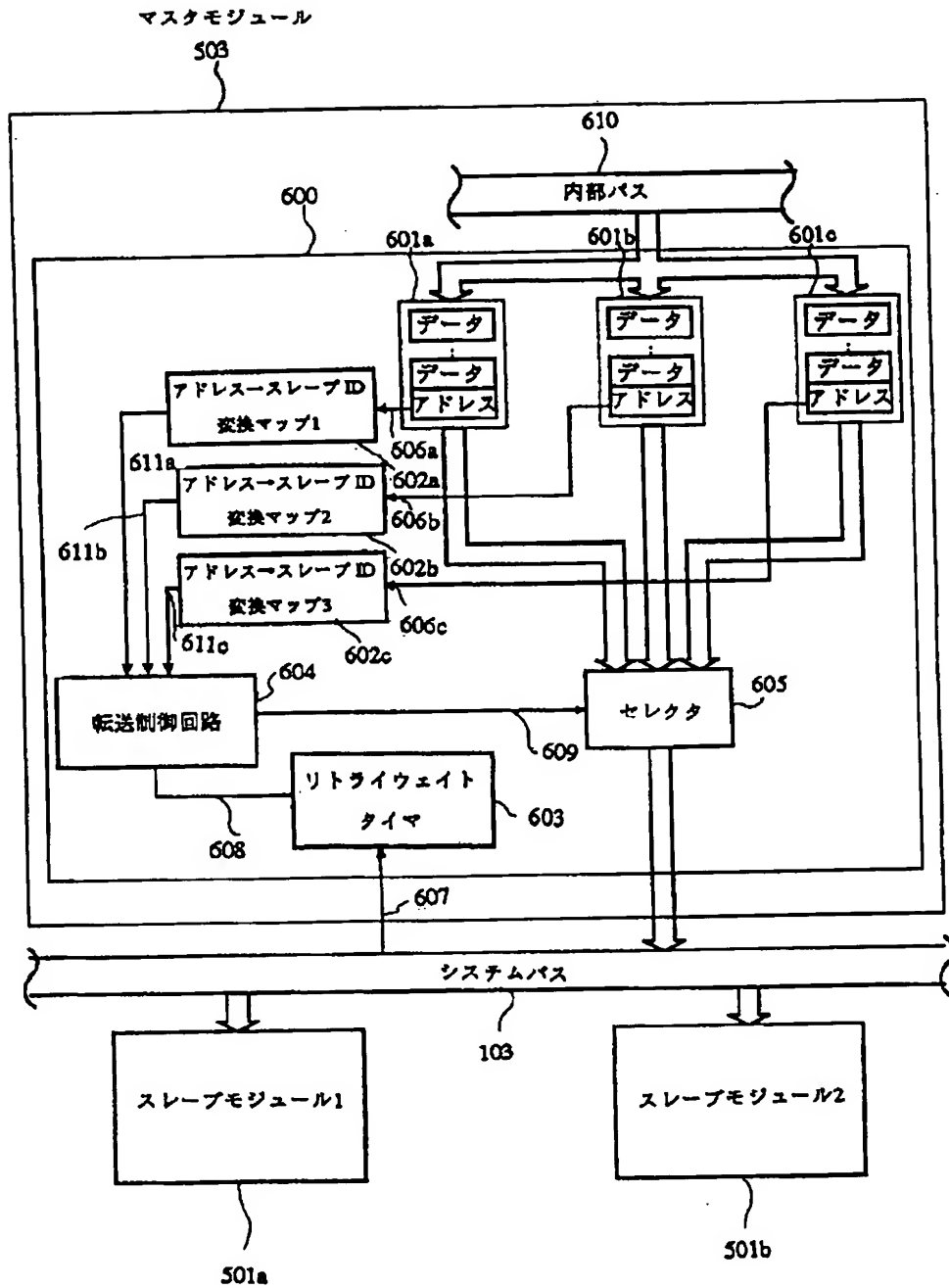
【図5】



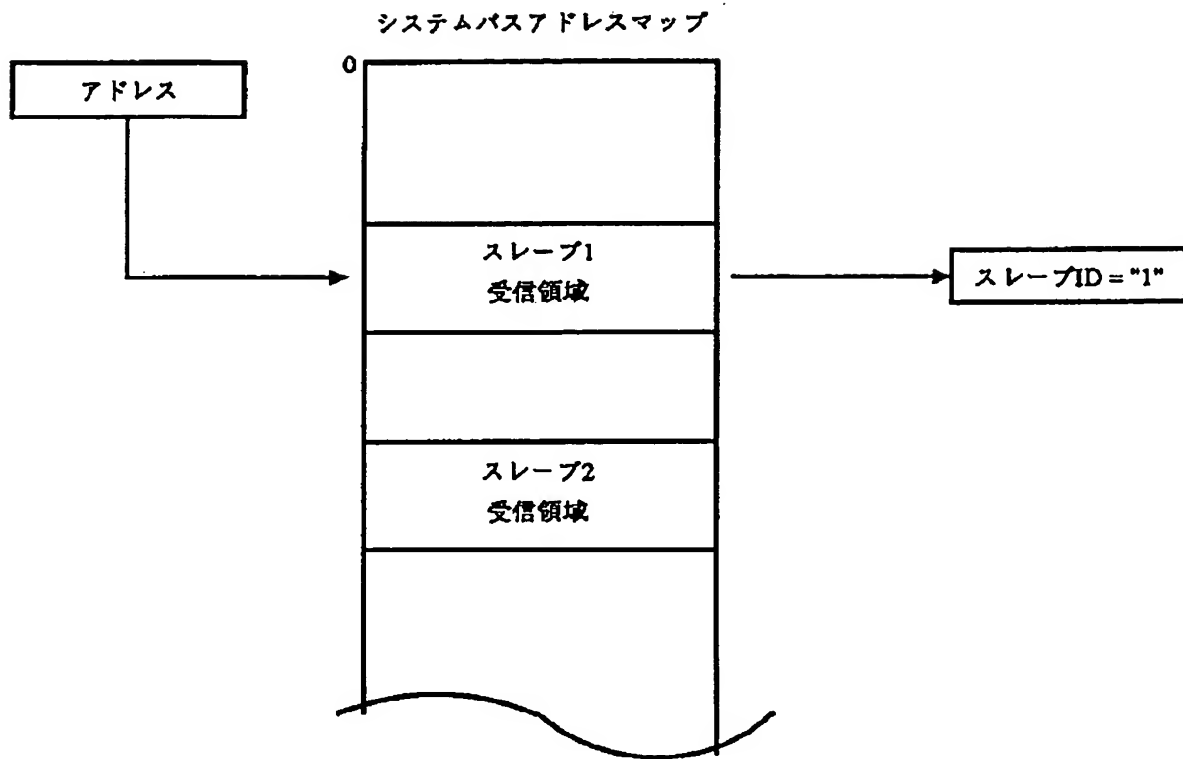
【図7】



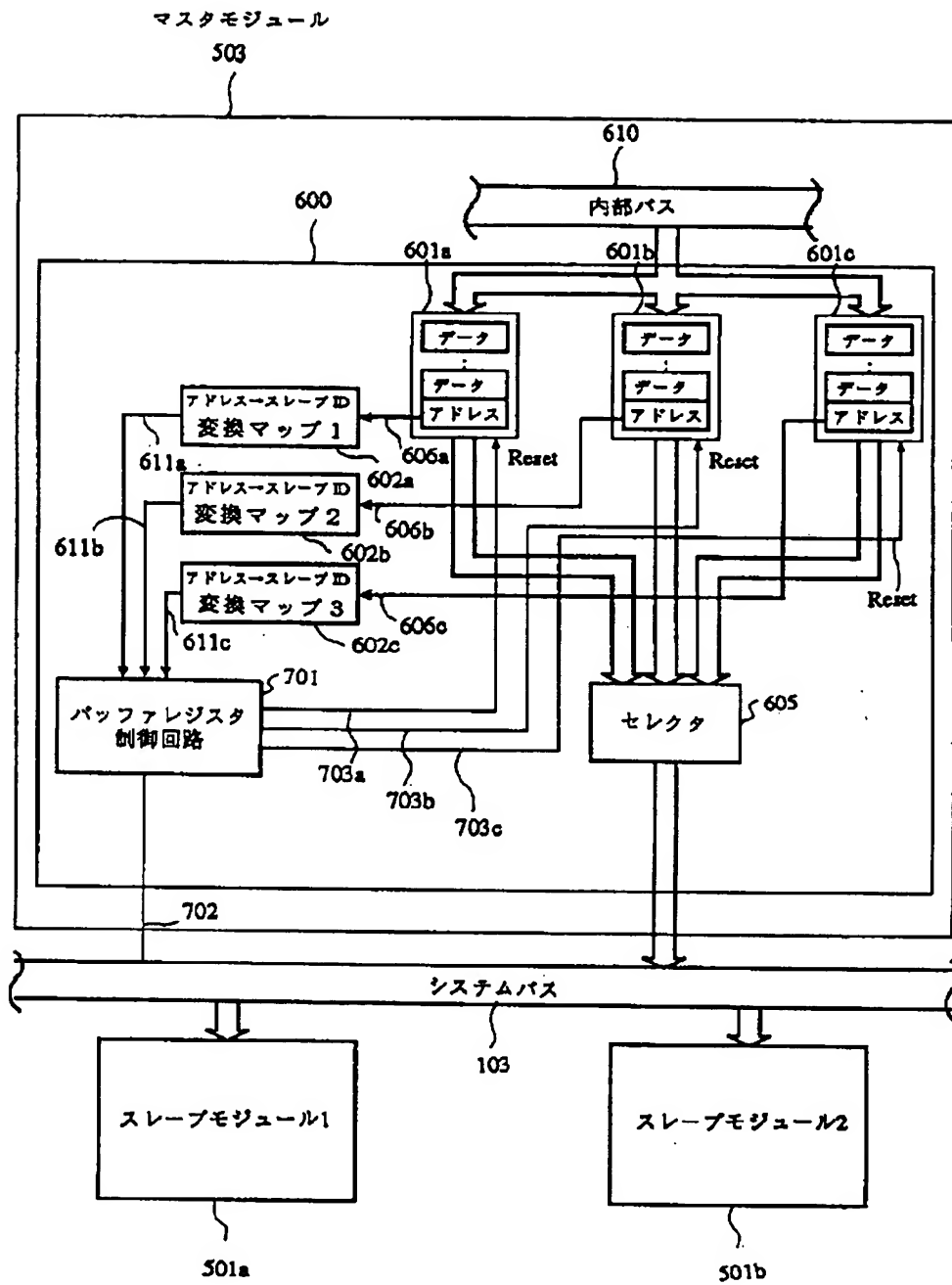
【図9】



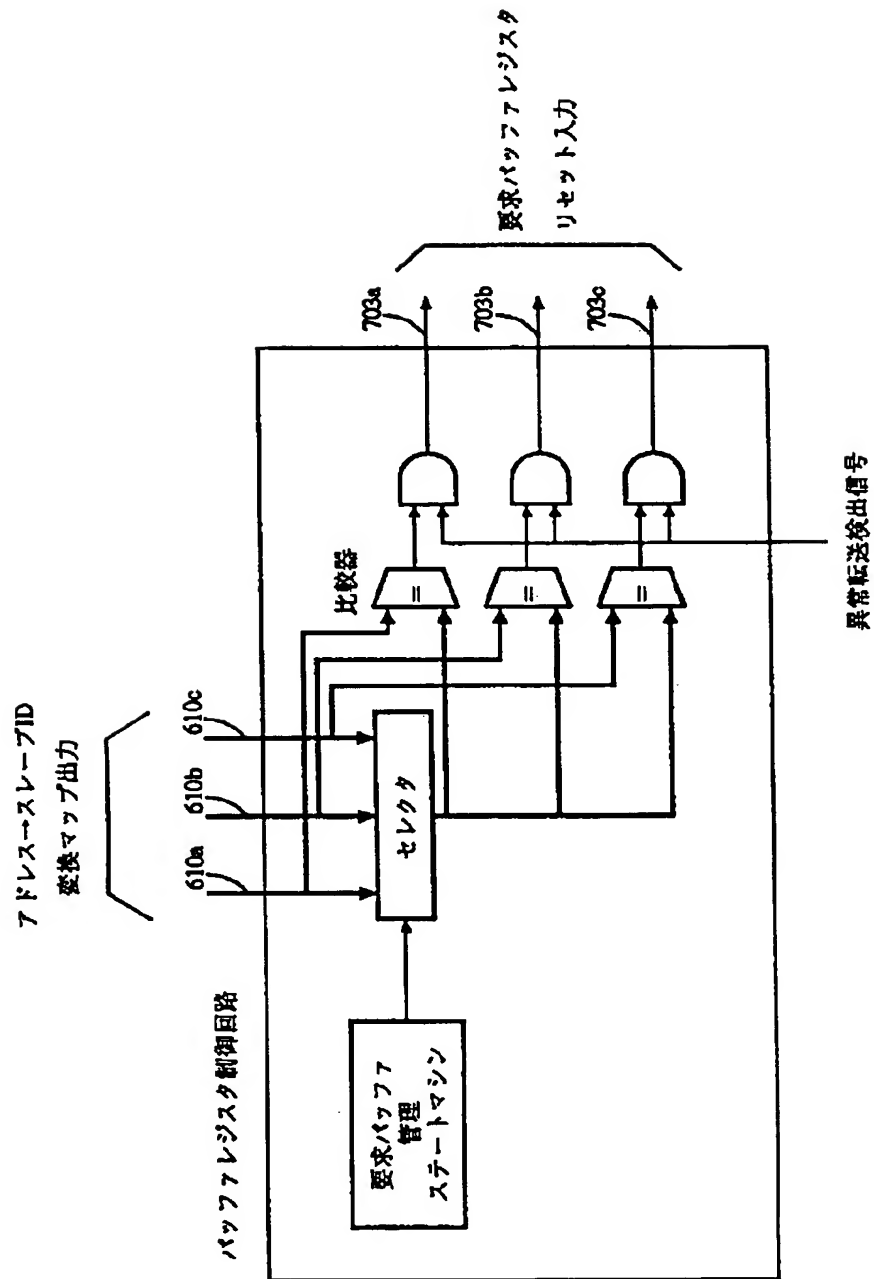
【図11】



【図12】

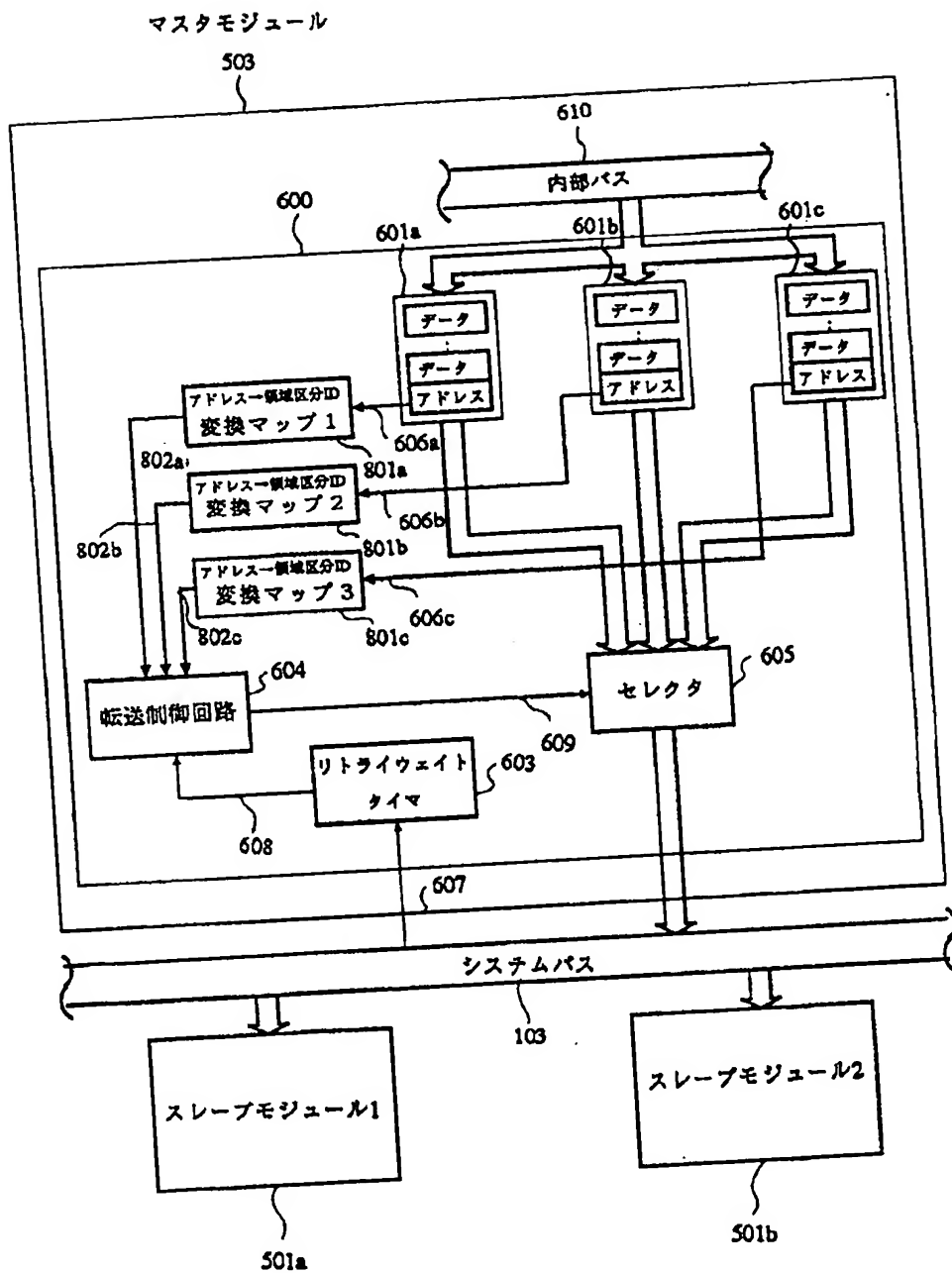


【図13】

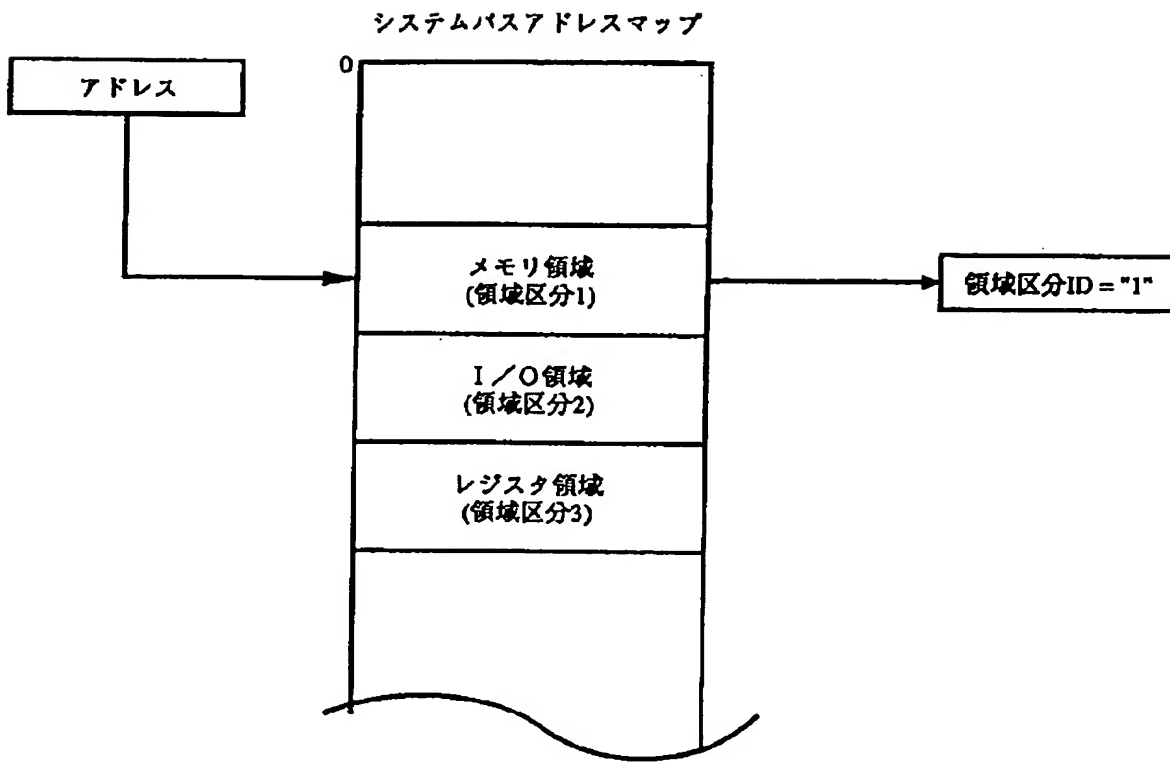


(22)

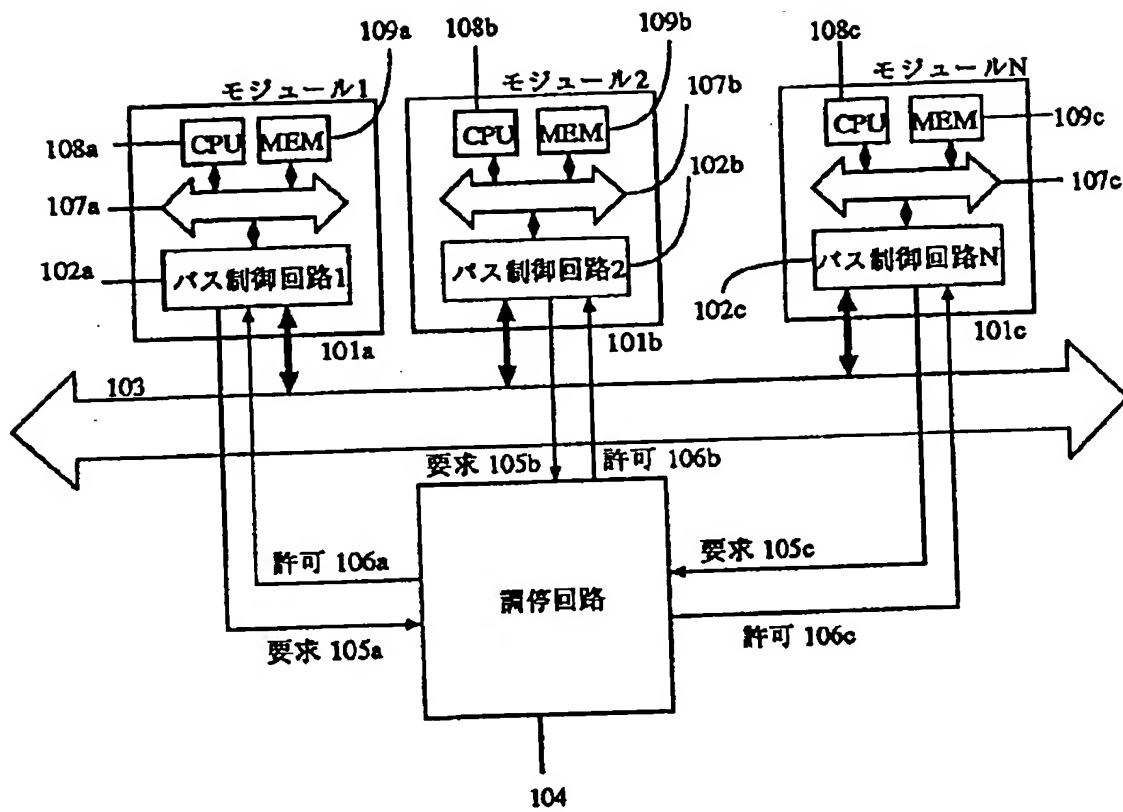
【図14】



【図15】



【図16】



The diagram illustrates a protocol control circuit (901) and its internal components. The circuit 901 is connected to an external protocol control circuit (902) and a protocol control circuit (903). It includes a protocol control circuit (904) and a protocol control circuit (905). The circuit 901 is connected to a protocol control circuit (902) and a protocol control circuit (903). It includes a protocol control circuit (904) and a protocol control circuit (905). The circuit 901 is connected to a protocol control circuit (902) and a protocol control circuit (903). It includes a protocol control circuit (904) and a protocol control circuit (905).